tion bit is high, the **1Y** output is at 12 Volts and the **2Y** output is at ground, causing current to flow through the magnet wire in one direction. When the direction bit is low, the **1Y** output is at ground and the **2Y** output is at 12 Volts, causing current to flow in the other direction. Flyback diodes **890** prevent the inductance of the magnet wire from damaging the motor driver chip when the outputs are switched off. Coil cable **114** couples actuator circuit **800** to magnet wire **108**. Motor driver chip is high-current driver chip. Suitable parts are available from Texas Instruments under trade designation L293D. Flyback diodes **890** are high-current fast-recovery diodes, such as the UF1002T made by Vishay Intertechnology of Malvern, Penn..

[0089] Actuator circuit **800** is just one way to interface a processor to an actuator. Many different circuits are possible, depending on the bandwidth and accuracy desired and the type of actuator being interfaced, among other factors. For example, a digital-to-analog converter could be used in combination with a linear amplifier to drive an electromagnetic actuator.

[0090] As a result of coil inductance, the relationship between the force generated by a given haptel and the width of the pulse generated by actuator circuit **800** may not be linear. Actuator circuits **800(1)**-(N) are therefore calibrated prior to use by measuring the force actually generated by each output value, and storing that measurement. Later, when a certain output force is desired, the correct output value to use is calculated by interpolating between the closest desired output values.

[0091] Substantial energy is dissipated by magnet wires **108(1)**-(N) at its peak force output. To minimize heat buildup in the haptel, coil holder **104** and surface **102** are preferably made of a material with high heat conductance. This allows heat to be conducted upward to surface **102**, where the heat can radiate away. Heat also radiates to stationary assembly **300**, which can be configured to act as a heat sink in combination with unified support plate **602**.

Control system operation

[0092] **FIG. 10** illustrates a flow diagram of the operation of I/O device **900** according to one embodiment of the present invention. The flow diagram of **FIG. 10** describes the actions performed in running control system **902** (e.g., by software). The management of I/O device **900** is preferably handled as a real-time task, so a separate control processor is preferably used instead of running control software on computer **916**. The control loop's frequency can be an important factor in the operation of I/O device **900**. The computational performance of control processor **904**, the speed of input and output operations, and the degree of optimization of the control software all affect the frequency (and thus responsiveness) attainable on a specific system. In one embodiment, a control loop frequency of 10 kHz is used.

[0093] The process begins with the reading of the haptels' position sensors (step **1002**). Analog input card **906** is preferably programmed during initialization to sample all inputs cyclically (e.g., for a system with nine haptels, and so, nine inputs, this would be at an aggregate frequency of 90 kHz providing a per haptel sampling rate of 10 kHz, and thus matching the control loop frequency). The sampling rate is preferably selected to exceed the bandwidth of proximity sensors **308(1)**-(N) and the maximum pass frequency of position circuits **700(1)**-(N) by a factor of two. The stored calibration values are used to compute the calibrated position value of each moving assembly (e.g., in millimeters). These calibrated position values may then be stored for later use.

[0094] Derived measurements are computed for each haptel using position values (step **1004**). The current velocity is computed by differentiating the position values. The current acceleration is computed by differentiating velocity values. In both cases, the result is low-pass filtered in order to reduce noise. The net force acting on moving assembly **100** is computed by dividing the acceleration of moving assembly **100** by the mass of moving assembly **100**. The force applied to XY sensor **116**, referred to herein as the applied force, is the result of subtracting the actuator force, the spring force and the weight of the moving assembly from the net force. The actuator force is known because the control computer sets the actuator force, the spring force is computed from the spring constant and the current position of moving assembly **100**, and the weight of the moving assembly was measured prior to assembly. Thus, the velocity, acceleration and applied force for each haptel can be computed from position measurements. Although derived measurements may have more noise, higher latency and less bandwidth than the original position measurements, such measurements are still adequate to allow implementation of the methods described herein.

[0095] Next, control processor **904** reads XY data from serial ports (step **1006**). XY interfaces **912(1)**-(N) send updated XY position values relatively infrequently when compared to the relatively high frequency at which the control loop runs, and so there is normally no data waiting. If data is waiting, control processor **904** reads the updated value and stores that value in memory. The value either encodes the location of a touch, or encodes that there is no touch applied to the sensor. The XY values are converted to a universal coordinate system using the position of the haptel within the haptel grid to offset the haptel's XY sensor reading.

[0096] Control processor **904** examines the incoming XY data for new touches (step **1008**). If a touch has appeared on a haptel since the last data point, three cases are possible. If this haptel is already part of a coordinated touch, no action is taken. If the haptel is not currently part of a coordinated touch, control processor **904** determines if a touch recently appeared within a predetermined distance of the haptel and within a predetermined time period. If such a touch is found, then this touch is presumed to be part of the same touch, and this haptel is added to the coordinated touch. If no such touch is found, then a new coordinated touch is started with this haptel as the only member.

[0097] Control processor **904** updates the collective measures for all coordinated touches (step **1010**). For each coordinated touch, the collective position is computed as an average of the vertical position values for each haptel belonging to the touch, preferably with each value weighted by the applied force for that haptel, as derived earlier (step **1004**). Thus haptels with no applied force do not contribute to the collective position measurement. Other weightings are possible, including equal weighting. Likewise, the collective velocity and collective acceleration are computed as a weighted average of these values for the haptels that con-